

REMARKS

Claims 1-39 are pending in the present application. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 103, Obviousness

The Office Action rejects claims 1-7, 10-16, 19-25, 28, 29, 33, 36 & 37 under 35 U.S.C. § 103 as being unpatentable over *Smith et al.* (US Patent No. 6,477,703 B1) in view of *Chamberlain* (US Patent No. 6,427,227 B1). This rejection is respectfully traversed.

With respect to claim 1, the Office Action states:

Regarding claims, 1 Smith discloses, a method for updating code, the method comprising: providing an update to a plurality of versions of a program, wherein the program is updated by an installer program and a plurality of versions of the installer program exist (Smith, fig. 1, 110, fig. 3, 330); determining whether a version of the installer program is incorrect with respect to the update (fig. 3, 355, 360); responsive to the version of the installer program being incorrect, updating the installer program from files in the update; and installing the update in the program with the updated installer program (fig. 5, 540). Smith doesn't explicitly disclose updating an installer program. However, Chamberlain does disclose updating the installer program (fig. 3, 31, 37). Therefore it would have been obvious to one of ordinary skills in the art at the time the invention was made to modify Smith with Chamberlain to attain the instant claimed invention because, "no software program is perfect and the release of numerous software patches to fix specific problems, and the release of new versions to fix or upgrade major problems are the norm" (Smith, 1:20-23).

Office Action, dated June 13, 2003. Applicant respectfully disagrees. *Smith* teaches a software patch selection tool providing a method and apparatus for identifying software patches for installation on a computer system in which current versions of software applications installed on a computer system are identified, as well as currently installed patches, products, and versions. An initial list of recommended patches is generated based on the combination of installed products and the validated list of datasheet selected products. See *Smith*, Abstract. The initial list of patches is checked for patches that are missing dependent patches and dependencies are added, as necessary. The initial list is also checked against itself and against a list of installed patches to remove redundant, obsolete, or outdated patches. Bad patches and patches that cause structural conflicts are

also removed. A user may then inspect the list and deselect unwanted patches. The user may also manually add other patches to the list. The tool then downloads the patches and orders them for execution. See *Smith*, col. 2, lines 17-41. *Smith* does not teach or suggest "updating the installer program from files in the update" and "installing the update in the program with the updated installer program," as recited in claim 1.

Chamberlain teaches a mechanism for repairing an installed and patched application program if a patched resource needed by the application program becomes inadvertently deleted or otherwise unavailable to the application program. An installer receives a patch and performs the patching operations necessary to apply the patch to the installed product. *Chamberlain* states:

The patching operations may include adding new program files to the installed product, altering existing program files associated with components of the product, modifying entries within the system registry 39, or other patching operations.

Chamberlain, col. 11, lines 7-11. Also, the installer updates an installer registry to reflect the existence of the patch and to reflect the proper installed state of the product after the patch has been applied. See *Chamberlain*, col. 10, lines 49-52.

The Office Action alleges that *Chamberlain* teaches updating the installer program in Figure 3, elements 31 and 37. This figure is shown below:

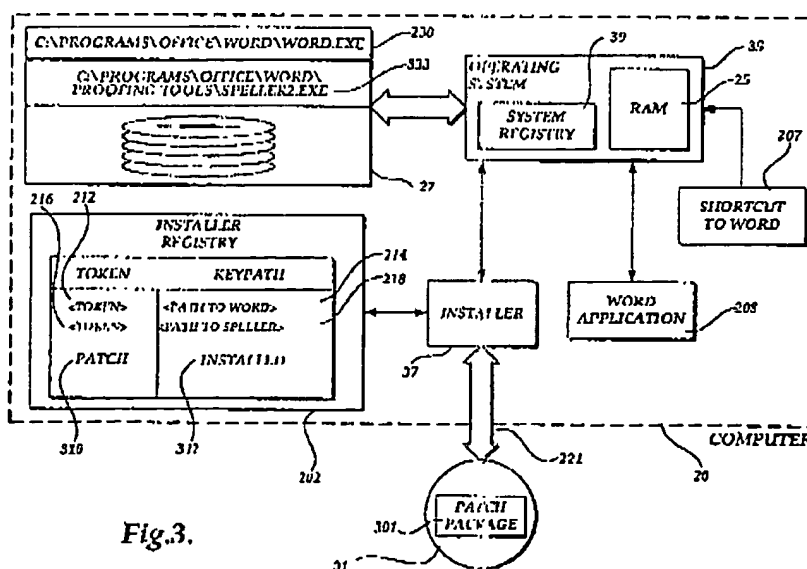


Fig.3.

Unfortunately, the reference number 37 does not appear anywhere in the text of *Chamberlain*. However, from the above portions, it is clear that the patch package does not update the installer program as alleged in the Office Action. Rather, the installer program 201 updates software in hard drive 27 using the patch package 301.

In contradistinction, the present invention provides a method for updating code in a software installer program. Claim 1 recites:

1. A method for updating code, the method comprising:
 - providing an update to a plurality of versions of a program, wherein the program is updated by an installer program and a plurality of versions of the installer program exist;
 - determining whether a version of the installer program is incorrect with respect to the update;
 - responsive to the version of the installer program being incorrect, updating the installer program from files in the update; and
 - installing the update in the program with the updated installer program.

Neither *Smith* nor *Chamberlain* teaches or suggests "determining whether a version of the installer program is incorrect with respect to the update," "responsive to the version of the installer program being incorrect, updating the installer program from files in the update," and "installing the update in the program with the updated installer program," as recited in claim 1. Since the applied references, taken alone or in combination, fail to teach or suggest every claim limitation, claim 1 cannot be rendered obvious by a combination of *Smith* and *Chamberlain*. Therefore, the Office Action fails to establish a *prima facie* case of obviousness and the rejection of claim 1 should be withdrawn.

Independent claims 10, 19, 28, 32, and 36 recite subject matter addressed above with respect to claim 1 and are allowable for the same reasons. Since claims 2-7, 11-16, 20-25, 29, 33, and 37 depend from claims 1, 10, 19, 28, 32, and 36, the same distinctions between *Smith* and *Chamberlain* and the invention recited in claims 1, 10, 19, 28, 32, and 36 apply for these claims. Additionally, claims 2-7, 11-16, 20-25, 29, 33, and 37 recite other additional combinations of features not suggested by the references, alone or in combination.

More particularly, with respect to claim 3, the Office action states:

Regarding claim 3 the method as recited in claim 1, wherein a determination that the version of the installer is more recent than the

update indicates that the version of the installer program is incorrect (5:15-20 for recent see recommended superseded by installed patch, and 50-55, also see user and deselecting conflicting patches 5:48-51).

Office Action, dated June 13, 2003. Applicant respectfully disagrees. The cited portion of *Smith* state:

All of the functionality of an older patch in a family tree is rolled into new patches in that family tree, thus, there is no reason to send an older patch, if a newer one is already on the system. At step 510, a decision is made as to whether there are any recommended patches superseded by installed patches. If so, the recommended patches are removed from recommended list (step 515).

Smith, col. 12-20. This portion of *Smith* actually teaches away from the invention recited in claim 3, because *Smith* teaches that a more recent version of a patch supersedes a less recent version. This is the opposite of the feature recited in claim 3, which recites "wherein a determination that the version of the installer is more recent than the update indicates that the version of the installer program is incorrect." The cited portion of *Smith* also states:

If common modules exist between patches that are dependent on each other, they can also be ignored as they're related although not in the same family tree (step 610). At step 620, any conflicts are displayed to allow the user to choose between the conflicting patches. The user can "click" on the desired patch (resulting in a deselect of the other patch(es) from the final master list). The Final Master Patch List is then displayed (Step 630). Each patch is listed on a separate line with a checkbox in front to allow the user to deselect (therefore not send) a patch.

Smith, col. 5, lines 46-55. Thus, *Smith* teaches that patches that depend on one another (newer versions may depend upon older versions) are ignored and, thus, not found to be incorrect. Also, the cited portion of *Smith* teaches that a user can decide between conflicting patches. However, there is no teaching in *Smith* of a determining step, wherein a newer version of an update is found to be incorrect. *Chamberlain* also fails to make up for the deficiencies of *Smith*.

The applied references, taken alone or in combination, fail to teach or suggest each and every claim limitation; therefore, claim 3 is cannot be rendered obvious by a

combination of *Smith* and *Chamberlain*. Claims 12 and 21 recite subject matter addressed above with respect to claim 3 and are allowable for the same reasons.

With respect to claim 4, the Office Action states:

Regarding claim 4 the method as recited in claim 1, wherein the version of the installer program is determined from a single one of a plurality of files contained within the installer program.

Office Action, dated June 13, 2003. Applicant respectfully disagrees. The Examiner bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). Therefore, Applicant respectfully requests, under 37 C.F.R. § 1.104(c), that the Examiner point out the particular part of *Smith* or *Chamberlain* relied on since *Smith* and *Chamberlain* show or describe inventions that are different from that claimed by Applicant. The Office Action merely repeats the claim limitations without any analysis as to why the claim is allegedly obvious. Thus, the Office Action fails to establish a *prima facie* case of obviousness. Applicant submits that *Smith* and *Chamberlain*, taken alone or in combination, fail to teach determining the version of an installer program from one single file contained within the installer program. Claims 13 and 22 recite subject matter addressed above with respect to claim 4 and are allowable for the same reasons.

With respect to claim 5, the Office Action states:

Regarding claim 5 the method as recited in claim 1, wherein the updating step comprises:

extracting installer files from the installer program into a directory (Chamberlain 10, 50-55);

overwriting selected files from the installer program with the corresponding updated file extracted from the update (Chamberlain, fig. 7, 705, 707, 709); and

packaging the updated files and remaining installer files into an updated installer program (Chamberlain, 10:49-53, see updating registry to reflect proper state after patching).

Office Action, dated June 13, 2003. Applicant respectfully disagrees. Figure 7 of *Chamberlain* is as shown below:

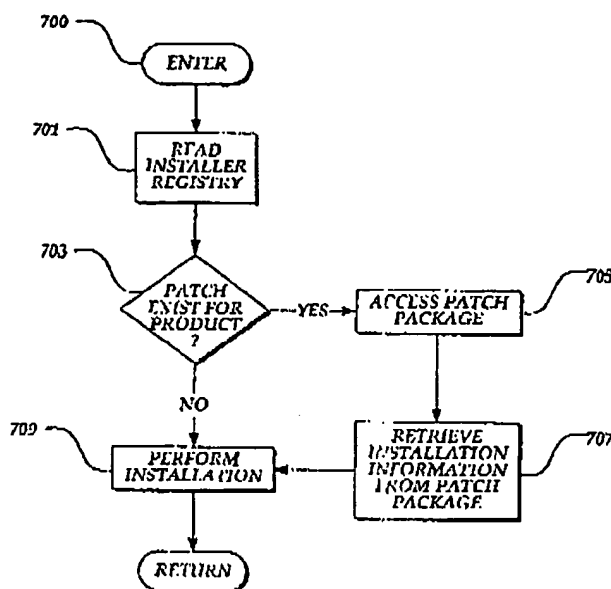


Fig.7.

Clearly, this figure does not teach or suggest “overwriting selected files from the installer program with a corresponding updated file extracted from the update,” as recited in claim 5. With respect to performing installation, *Chamberlain* states:

At step 709, the installer application 201 performs an installation (or reinstallation) of the missing file to the location identified in the installer registry 202. In this example, the installation may include reading the original key file speller.exe 230 (FIG. 2) from the source 215, altering that program file with the patch bits from the patch package 301 to create the patched key file speller2.exe 333, and writing the patched key file speller2.exe 333 to the location identified by the keypath 218. It will be appreciated that reinstalling the missing program file may possibly include reinstalling other program files associated with the missing program file. In one example, the installer application 201 may install components of the product rather than individual program files, with each component having more than one associated program file.

Chamberlain, col. 17, lines 52-63. Clearly, this step performs installation, using the installer application, to an installed application, rather than to the installer application itself.

Furthermore, claim 5 recites, "packaging the updated files and remaining installer files into an updated installer program." The Office Action alleges that this step is taught by *Chamberlain*. The cited portion of *Chamberlain* states:

At step 414, the installer application 201 updates the installer registry 202 to reflect the existence of the patch, and to reflect the proper installed state of the product after the patch has been applied. For example, the installer application 201 may add a patch entry 310 in the installer registry 202 indicating the existence of the patch.

Chamberlain, col. 10, lines 49-53. The Office Action proffers no reasoning as to why updating a registry is equivalent to the claimed step of packaging updated files and remaining installer files into an updated installer program. Applicant submits that *Smith* and *Chamberlain*, taken alone or in combination, fail to teach or suggest this feature. Therefore, a combination of *Smith* and *Chamberlain* would not result in the invention recited in claim 5. Claims 14, 23, 29, 33, and 37 recite subject matter addressed above with respect to claim 5 and are allowable for the same reasons. Also, claims 6, 15, and 24 depend from claims 5, 14, and 23, respectively, and are allowable by virtue of their dependency.

More particularly, with respect to claim 6, the Office Action states:

Regarding claim 6 the method as recited in claim 5, wherein the packaging step comprises the updated files and remaining installer files to produce an updated installer program (*Chamberlain* 1:22-24).

Office Action, dated June 13, 2003. Applicant respectfully disagrees. The cited portion of *Chamberlain* states:

Often the program files are stored in a compressed format to conserve storage space.

Chamberlain, col. 1, lines 22-24. While compression of program files is generally known, the prior art as a whole fails to teach or suggest extracting (decompressing) files from an installer program, updating files from the installer program, and packaging and compressing the updated and remaining installer files to produce an updated installer

program, as recited in claim 6. Even though *Chamberlain* teaches compression of program files in general, the combination of *Smith* and *Chamberlain* fails to arrive at the claimed invention. Therefore, claim 6 is not rendered obvious by the proposed combination of *Smith* and *Chamberlain*. Claims 15 and 24 recite subject matter addressed above with respect to claim 6 and are allowable for the same reasons.

Therefore, the rejection of claims 1-7, 10-16, 19-25, 28, 29, 33, 36 & 37 under 35 U.S.C. § 103 is overcome.

The Office Action rejects claims 8, 9, 17, 18, 26, 27, 30, 31, 34, 35, 38, and 39 under 35 U.S.C. § 103 as being unpatentable over *Smith* in view of *Chamberlain* and further in view of *Forbes et al.* (US Patent No. 6,381,742). This rejection is respectfully traversed.

Claims 8, 9, 17, 18, 26, 27, 30, 31, 34, 35, 38, and 39 depend from claims 1, 10, 19, 28, 32, and 36 and are allowable by virtue of their dependency. While *Forbes* does teach that software packages are frequently written in common programming languages, such as Java, *Forbes* does not make up for the deficiencies of *Smith* and *Chamberlain*. More specifically, *Forbes*, like *Smith* and *Chamberlain*, fails to teach or suggest updating an installer program from files in an update, and installing an update in a program using the updated installer program. Therefore, since the applied references, taken alone or in combination, fail to teach or suggest each and every claim limitation, claims 8, 9, 17, 18, 26, 27, 30, 31, 34, 35, 38, and 39 cannot be rendered obvious by the proposed combination of *Smith*, *Chamberlain*, and *Forbes*.

Therefore, the rejection of claims 8, 9, 17, 18, 26, 27, 30, 31, 34, 35, 38, and 39 under 35 U.S.C. § 103 is overcome.

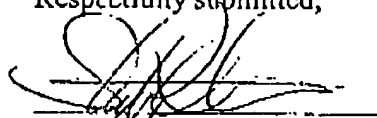
II. Conclusion

It is respectfully urged that the subject application is patentable over the cited prior art and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: Sep 15, 2003

Respectfully submitted,



Stephen R. Tkacs
Reg. No. 46,430
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Agent for Applicants

OFFICIAL**RECEIVED
CENTRAL FAX CENTER**

SEP 15 2003